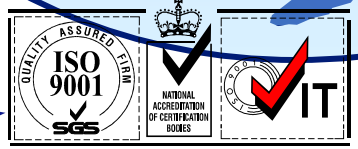
Exercise Clock

ISC2

Bruce Long
ISC2 Chief Architect
4 Apr 2002



® CMM is a registered trademark of Carnegie Mellon University

Agenda

- Clock Timer Implementation
- Public APIs
- User Interface
- Integration

Introduction

ISC2 Has a Requirement to Run Exercises at other than Real-time

- **Up to 10-1 Compression, Start, Stop, Pause, and Restart Required**

Required to Allow The System to Process Tracks Correctly From Sources That Are Running at Other Than Real Time

- **Allows a System to Operate at Offsets From Current Time, at Varying Compression Ratios Without Having to Reset the System Clocks on Servers and Clients**

Provides Critical Support to Testing, Exercises, and the Switch-over From Exercise to Real-world TMS Servers In the Event of Conditions Requiring Auto-Cutover

- **Eliminates the Need to Reset System Clocks on Switchover**

Exercise Clock Runs Forward Only.

Processing Occurs in Real-World Time only

- **Events May Occur in Fast or Slow Time, and Time-based Decision Are Based on Exercise Clock Behavior**
- **Processing in Progress At the Time of Stop or Pause Continues to Completion**

Clock Timer Implementation

Clock control:

- **Public APIs**
- **User interface**

Exercise Time Model:

- **Tsys = System Time (at workstation)**
- **Tstart = Start of exercise**
- **Rate = Acceleration factor**
- **dT = Delta Time (Tsys – Tstart) (.1 to 6 Times Wall Clock Rate)**
- **Exercise Time (Tout) = Rate*dT + Tstart**

Algorithm

- **At start, Enter Tstart, dT, Starts Exercise Clock**
- **When Exercise Clock is Running, Time = Tout (Rate*dT + Tstart)**
- **At Pause, Set Tpause = Tout**
- **At Resume, Set Tstart = Tpause**

Tstart and Rate, Clock Run Control Are Entered Via User Interface

Clock Control Layer

Maintains Independent System Time:

- Start/Pause/Resume/Stop Functions Through UI or Public APIs
- Exercise Start Time is Maintained Relative to Current System Time
- Selected Acceleration Rate Used to Adjust Duration of Exercise Time Interval

Non-volatile Storage Used For:

- User Interface Inputs, and
- Clock Parameters

Public APIs

Apis Utilized by User Interface Reusable by Arbitrary Java Program

API List:

- **Void startIflClock(long Tstart, double Rate)**
- **Void pauseIflClock()**
- **Void resumeIflClock()**
- **Void stopIflClock()**
- **Long getIflTime()**
- **Long getDTG()**
- **Void setClockMode(int Mode)**
- **Void setClockRate(double Rate)**
- **String[ ] getClockSettings()**

ICSF User Interface Exercise Mode

Integration

TMS Server – UNIX platforms only

Clients: All platforms including:

- **UNIX: Sun and HP**
- **Windows: NT and 2000**

Development and Runtime Environments

Status

ISC2 is Funding the Development and Integration into the ISC2

Once Implemented, The Change Will Roll Into COE Baseline

- **COE Program Office Agreed to Sustain It After Implementation**

Target Is to Create A Capability that Could Be Leveraged by Later by HLA-Compliant Simulations to Provide Clock Management

- **ISC2 Will Manage the ICSF Clock from the Public APIs, and Provide a System Level User (Vs ICSF Only) Interface to the User**

Several Other Programs have Expressed Interest in This Capability Once Available

ICSF Update is Still in Work. Delivery to ISC2 is Planned for NLT 1 May